Arjun Bagade - 2023300008

Aanish Bangre - 2023300011

TE COMP - Div A Batch A

SE LAB

Experiment 2: P2P Rental UML Diagrams

Use Case Diagram:

Table 1: List of Actors

Actor	Description
User (Renter)	A registered user who browses available items, makes bookings, pays rental charges, and communicates with owners. Renters can view listings, chat, receive notifications, and manage their profile.
User (Owner/ Lister)	A registered user who publishes rental listings, sets prices, availability, and security deposits. Owners manage their bookings, receive payments, and communicate with renters.

Table 2: List of Use-Cases

Sr No.	Use Case	Description
1	User Registration	Allows new users to create an account with email, password, and name.
2	User Login	Enables existing users to log in using registered credentials.
3	Session Management	Maintains JWT-based authentication and secure cookie storage.
4	Profile Setup	Users complete profile with username and phone number.
5	Profile Management	View, edit, and update personal profile details.
6	Reputation Tracking	Tracks user reliability scores for trust building.
7	Create Listing	Publish rental items with title, description, images, category, deposit, price, and location.
8	View Listings	Browse available items listed by other users.
9	Listing Details	Access complete information about an item.
10	Booking Creation	Reserve items for specific dates with calendar selection.
11	Payment Processing	Securely handle payments and deposits in INR.
12	Chat Creation	nitiate chat between renter and owner.
13	Real-time Messaging	Exchange instant messages in booking-based chat rooms.
14	Notifications	Generate and deliver system and chat notifications.

Use Case Scenarios:

Table 3: UC1 – User Registration

<table>
<tr><td>Use Case:</td><td>UC1. User Registration</td></tr>
<tr><td>Goal:</td><td>To create a new account in the application</td></tr>
<tr><td>Actors:</td><td>User</td></tr>
<tr><td>Pre-condition:</td><td>User is not registered in the system</td></tr>
<tr><td>Post-condition:</td><td>A new account is created and the user is logged in</td></tr>
<tr><td>Mainline Scenario:</td><td>
<table>
<tr><th>Actor Actions</th><th>System Actions</th></tr>
<tr><td>1. User accesses the application.</td><td></td></tr>
<tr><td>2. User selects “Sign Up” option.</td><td></td></tr>
<tr><td>3. User provides name, email, and password.</td><td>4. System validates the provided information.</td></tr>
<tr><td></td><td>5. If valid, creates new user account and generates session.</td></tr>
<tr><td></td><td>6. Logs in the user and redirects to profile setup.</td></tr>
</table>
</td></tr>
<tr><td>Alternate Flows:</td><td>3. If email already exists, system prompts user to choose another.</td></tr>
</table>

Table 4: UC2 – Create Listing

<table>
<tr><td>Use Case:</td><td>UC2. Create Listing</td></tr>
<tr><td>Goal:</td><td>To publish a rental item in the marketplace</td></tr>
<tr><td>Actors:</td><td>User</td></tr>
<tr><td>Pre-condition:</td><td>User is authenticated and profile is set up</td></tr>
<tr><td>Post-condition:</td><td>A new item listing is created and available for rent</td></tr>
<tr><td>Mainline Scenario:</td><td>
<table>
<tr><th>Actor Actions</th><th>System Actions</th></tr>
<tr><td>1.User selects “Publish Item” option.</td><td></td></tr>
<tr><td>2. User enters item details (title, description, price, deposit, availability, location, images).</td><td>3. Validates all entered information.</td></tr>
<tr><td></td><td>4. Stores item in database.</td></tr>
<tr><td></td><td>5. Sets item status to “Active” and publishes listing.</td></tr>
</table>
</td></tr>
<tr><td>Alternate Flows:</td><td>3. If required details are missing, system prompts user to fill them.</td></tr>
</table>

Table 5: UC3 – Booking Creation

Use Case:	UC3. Booking Creation	
Goal:	To reserve an item for specific dates	
Actors:	User	
Pre-condition:	User is logged in and listing is active	
Post-condition:	Booking record is created and owner notified	
Mainline Scenario:	Actor Actions	System Actions
	1. User selects item from listings.	
	2. User chooses start and end dates.	3. Displays total price to user.
	3. User confirms booking.	4. Creates booking record in database.
		5. Sends notification to item owner.
Alternate Flows:	4. If selected dates overlap with existing booking, system shows error.	

Table 6: UC4 – Payment Processing

Use Case:	UC4. Payment Processing	
Goal:	To securely process payments for bookings	
Actors:	User, Razorpay Gateway	
Pre-condition:	User has confirmed a booking	
Post-condition:	Payment is processed, verified, and receipt is generated	
Mainline Scenario:	Actor Actions	System Actions
	1. User proceeds to payment.	2. Generates Razorpay order linked with booking.
	2. User enters payment details.	4. Processes transaction through Razorpay.
		5. Verifies payment signature and confirms success.
		6. Issues receipt and updates booking as “Paid.”

Alternate Flows:	3. If transaction fails, system notifies user and prompts retry.

Table 7: UC5 – Chat Creation & Messaging

Use Case:	UC5. Chat Creation & Messaging
Goal:	To enable communication between renter and owner
Actors:	User (Renter), User (Owner)
Pre-condition:	Booking exists between users
Post-condition:	Chat session is created and messages are exchanged
Mainline Scenario:	Actor Actions / System Actions 1. Renter selects booking and opens chat. / 2. Creates chat room linked to booking. 2. Renter sends a message. / 3. Stores message in database and delivers in real-time. 3. Owner replies. / 4. Broadcasts message to renter. 5. Maintains chat history.
Alternate Flows:	None

Table 8: UC6 – Notifications

Use Case:	UC6. Notifications
Goal:	To alert users about booking, payments, and chat updates
Actors:	User
Pre-condition:	User is logged in
Post-condition:	User receives relevant real-time notifications
Mainline Scenario:	Actor Actions / System Actions 1. User performs or triggers activity (booking, chat, payment). / 2. System creates notification entry. 3. Pushes notification in real-time to target user.

	2. User opens notifications tab.	4. Displays latest notifications with type and timestamp.
Alternate Flows:	If no notifications exist, system shows “No new notifications.”	

Use Case Diagram: